Universität Stuttgart

www.visus.uni-stuttgart.de

Masterarbeit

Multi-timescale Dynamic Graph Visualization

Student:

Moataz Abdelaal

Supervisor:

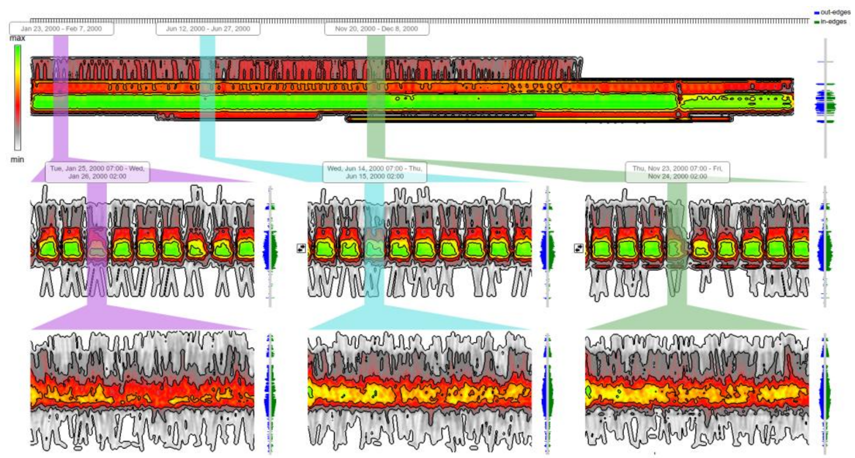
Dr. Michael Burch

Examiner:

Prof. Dr. Daniel Weiskopf

Overview

- **Problem:** how to provide an overview on multiple time scales of the time-varying relational data
- **Proposed Solution:** a multi-timescale dynamic graph visualization approach that is visually scalable in both dimensions, *time* and *edges*.
- **Contributions**
  - Compact visualization design (top-down and side-by-side stacking)
  - Reducing the amount of visual clutter (clustering and ordering techniques)
  - Set of interaction techniques
- **An application example:** to demonstrate the applicability of our approach



A multi-timescale view on a dynamic graph dataset acquired from the US domestic flight database

www.visus.uni-stuttgart.de

www.visus.uni-stuttgart.de

Outline

- Related work
- Multi-timescale visualization approach
- Vertex clustering and reordering
- Interaction techniques
- Application example
- Discussion and limitations
- Conclusions

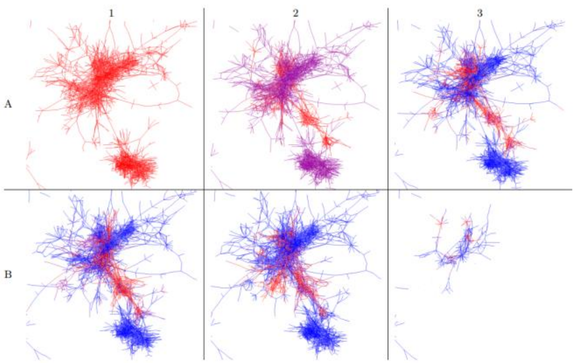
www.visus.uni-stuttgart.de

Related Work

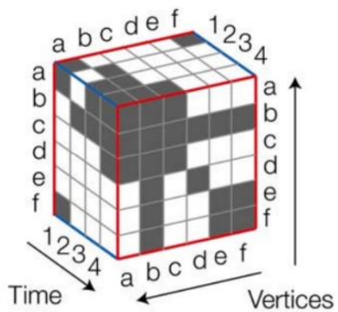
- Node-link based
  - **Gevol**
  - − Limited scalability with large graphs
  - − Space-inefficient

- Matrix based
  - **Cubix**
  - **Small MultiPiles**
  - + Compact visual representation
  - − Limited scalability w.r.t. time steps
  - − Multiple time scales not supported
  - − Comparison tasks not supported

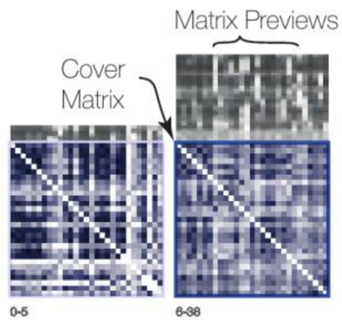
- Radial-layout based
  - **TimeRadarTrees**
  - **Circular MSV**
  - − Much screen space is needed
  - − Comparisons of time moments are harder



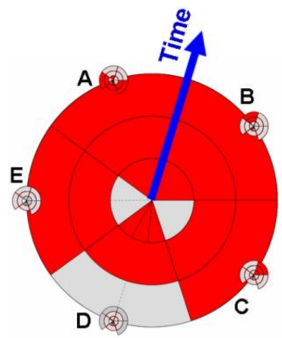
Gevol (Collberg et al. 2003)



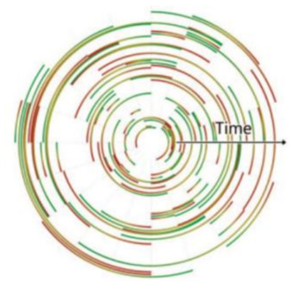
Cubix (Bach et al. 2014)



MultiPiles (Bach et al. 2015)



TimeRadarTrees (Burch et al. 2008)



Circular MSV (Van den Elzen et al. 2014)

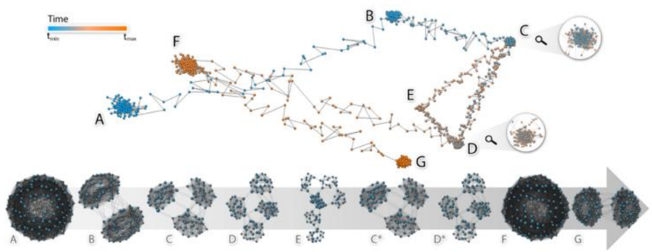
www.visus.uni-stuttgart.de

Related Work

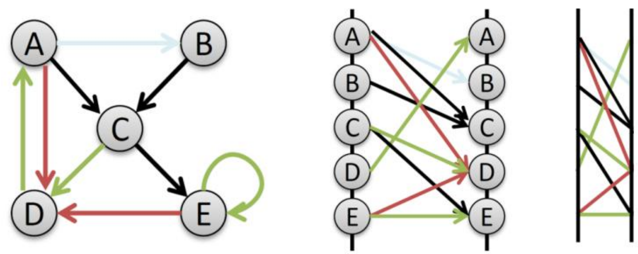
- **Reducing graphs to points**
  - Graphs are considered points in high-dimensional space
  - − Scalability concerns w.r.t. graph size and number of time steps
  - − Harder to interpret the resulting dimensions

- **Parallel Edge Splatting**
  - Splatting is employed to address overplotting of edges
  - \+ Visual scalable
  - − Scalability concerns w.r.t. the number of time steps

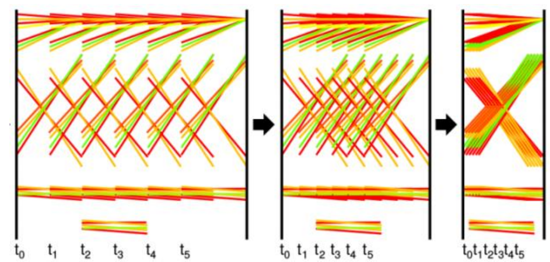
- **Visualizing a Sequence of a Thousand Graphs**
  - Interleaving the vertical stripes together with only one pixel offset
  - \+ Scalable w.r.t. the number of time steps
  - \+ Emphasizes the graph structures



Reducing Snapshots to Points (Van den Elzen et al. 2016)



Parallel Edge Splatting (Burch et al. 2011)



Visualizing a Sequence of a Thousand Graphs (Burch et al. 2017)

Universität Stuttgart

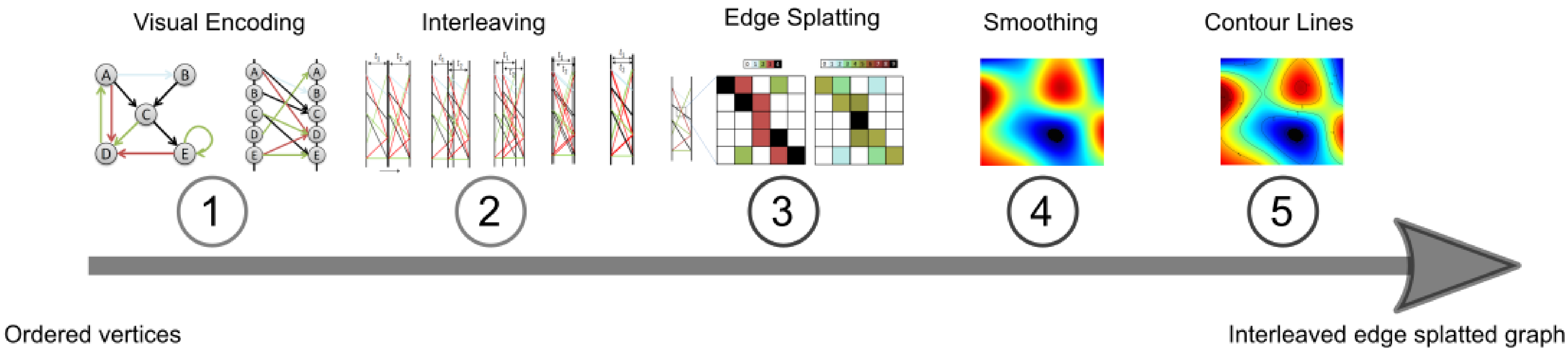
www.visus.uni-stuttgart.de

Our Contribution

- Extending *Burch et al. 2017* visualization approach by:
  - Showing the dynamic graph structures on multiple time scales directly in a combined fashion
  - Reducing the amount of visual clutter through clustering and ordering techniques
  - Providing a set of interaction techniques

Visualizing a Sequence of a Thousand Graphs (Burch et al. 2017)

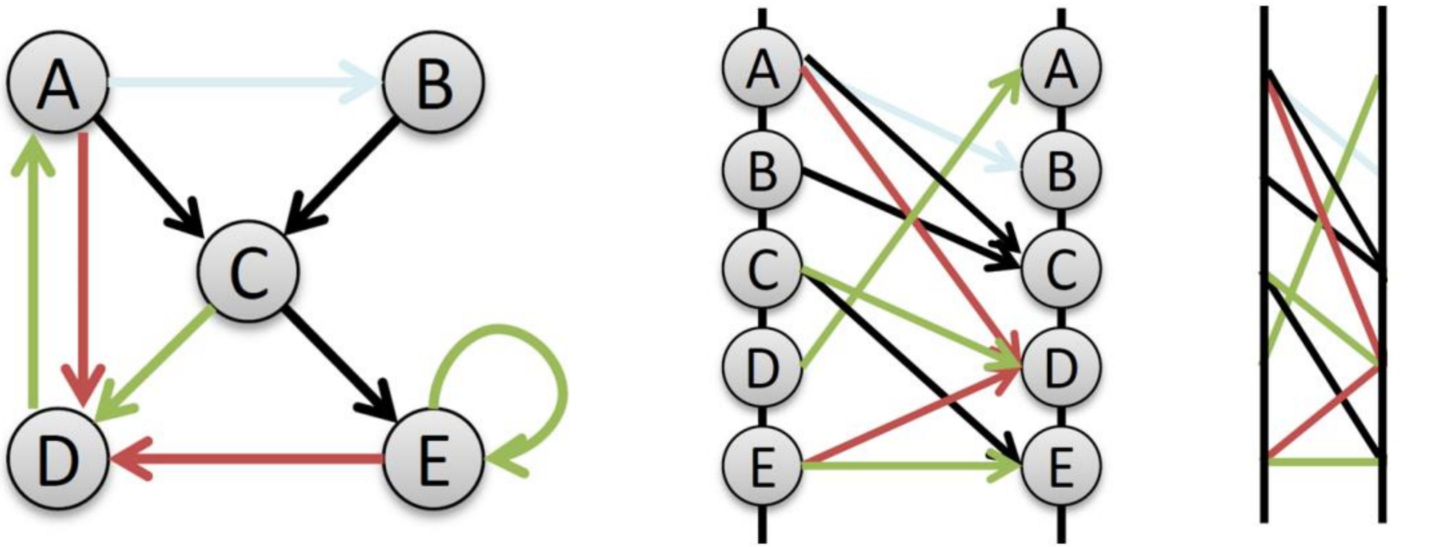
- Time-to-space mapping
- Providing an *overview* of the evolution of graph dynamics over time
- Individual graph sequences go through a visual mapping pipeline



A visual mapping pipeline

Encoding of Individual Time Steps

- The transformation of the graph into a bipartite representation
- Each time step is represented by two vertical, parallel lines (stripe)
- Vertices are placed on both lines with the same order
- The edge direction is encoded in the left-to-right reading direction



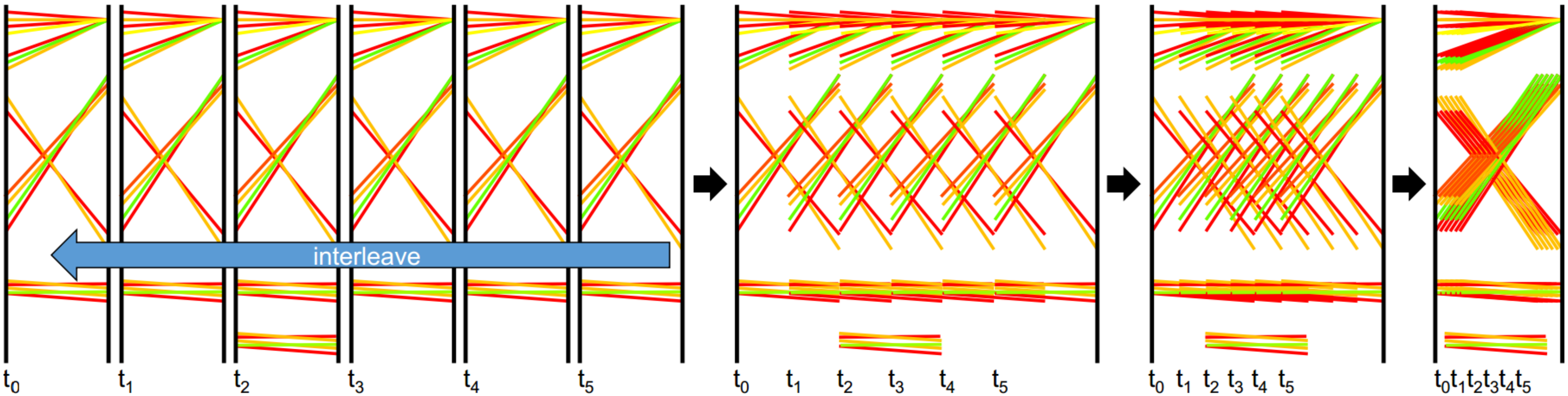
The transformation of the graph into a bipartite representation (Burch et al. 2017)

Universität Stuttgart

www.visus.uni-stuttgart.de

Interleaving

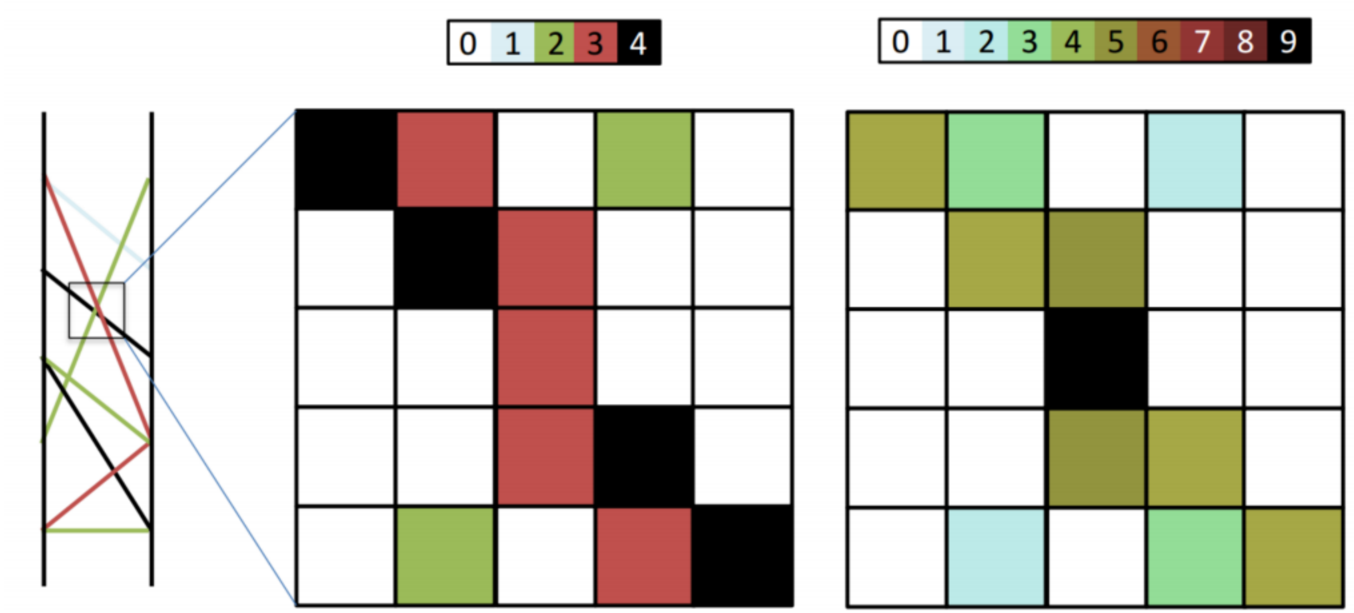
- Let the stripes overlap with only one pixel offset
- The thickness of the visible structures is an indicator for the time period



Interleaving (Burch et al. 2017)

Edge Splatting (Burch et al. 2011)

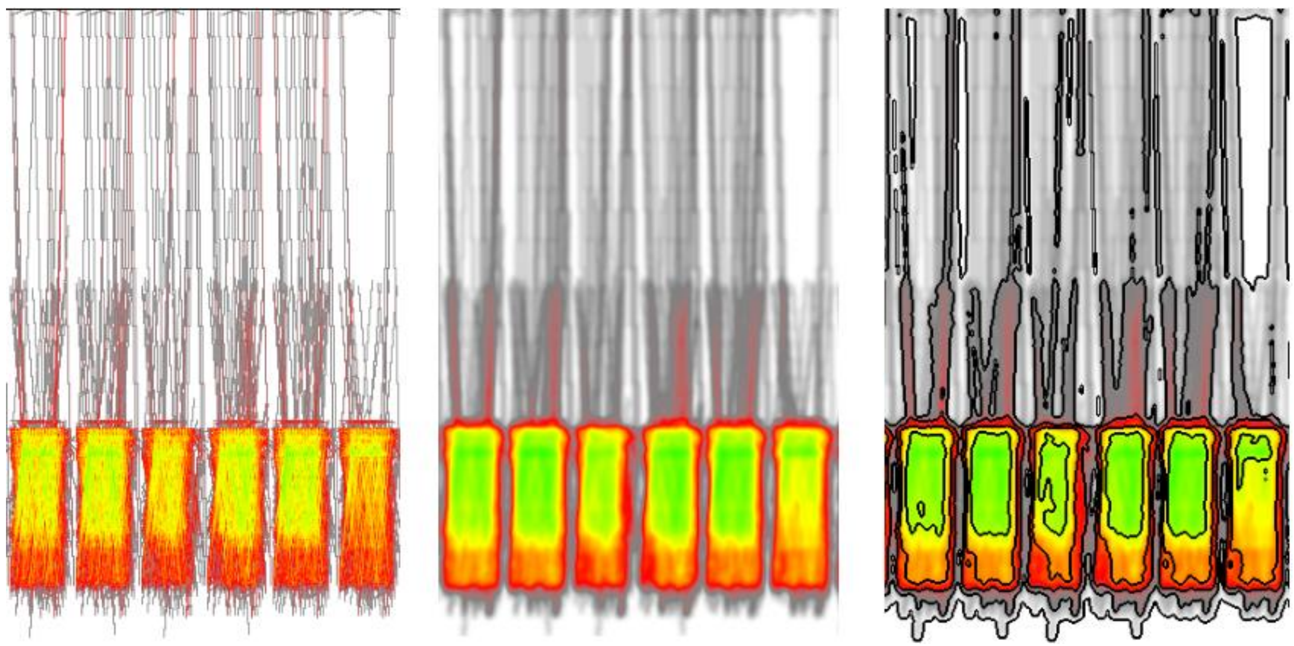
- To produce a less cluttered visual representation
- A scalar density field that aggregates the overlapping edges is computed and displayed.



Edge Splatting (Burch et al. 2011)

Smoothing and Contour Lines

- To obtain a less cluttered visual representation
- Apply a low pass filter several times to produce a smoother result
- Augmenting black contour lines
  - Our brains designed to seek out continuous contours [War04]



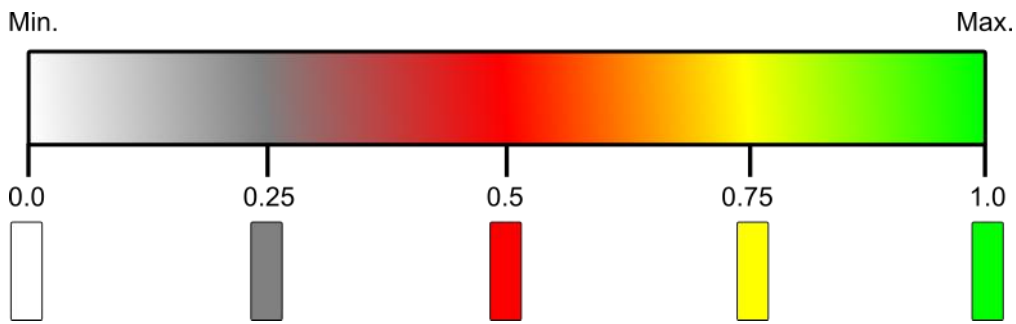
Before smoothing *After smoothing* *Contour lines augmented*

[3×3] box filter applied 5 times

www.visus.uni-stuttgart.de

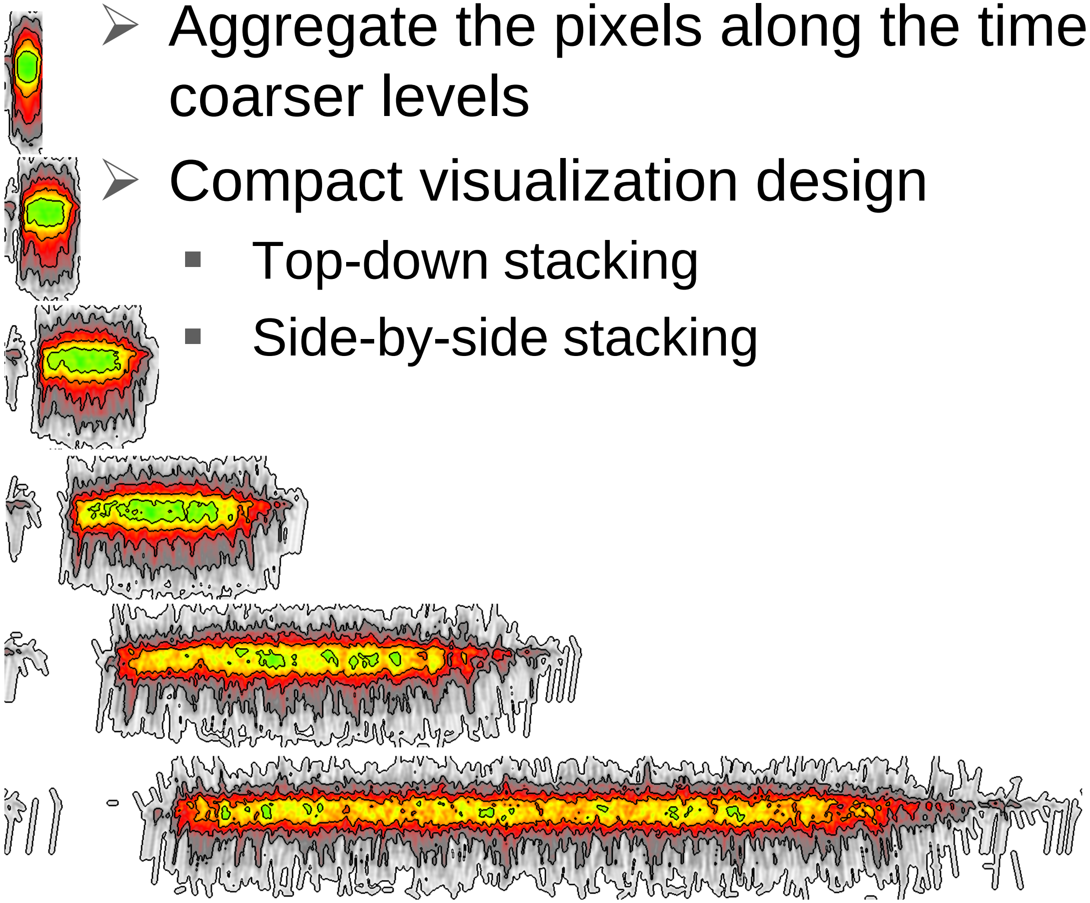
Color Mapping

- A logarithmic function is applied to the pixels scalar weights
- Logarithmic weights are then normalized on a scale of 0 to 1
- Linear interpolation is done to approximate the final pixel color



Discrete color bar with five color bins

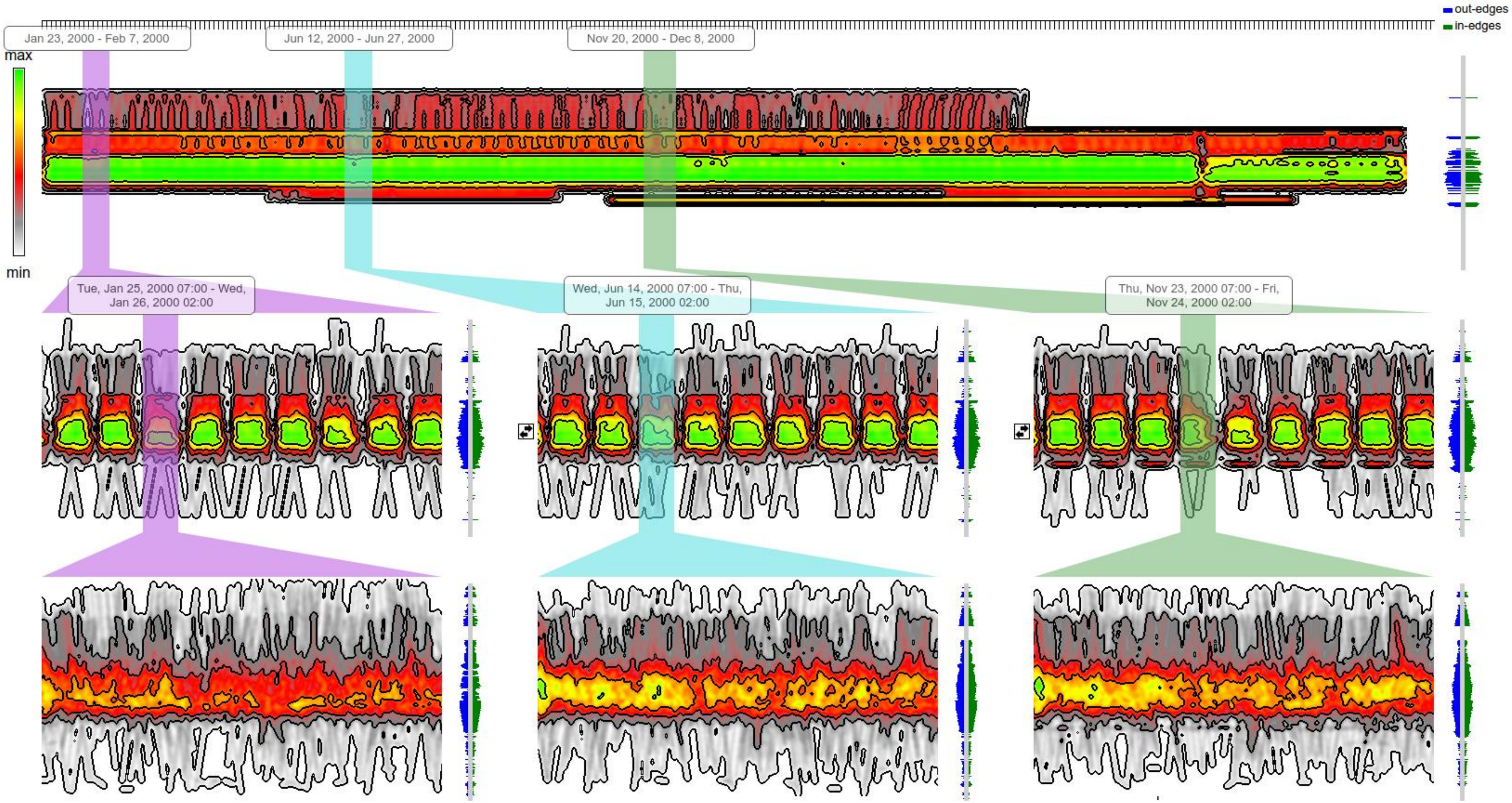
Visualizing on Multiple Time Scales



- Aggregate the pixels along the time-axis to generate the coarser levels
- Compact visualization design
  - Top-down stacking
  - Side-by-side stacking

Aggregate the pixels along the x-axis to generate the coarser levels

Visualizing on Multiple Time Scales



Compact visualization design: top-down stacking and side-by-side stacking

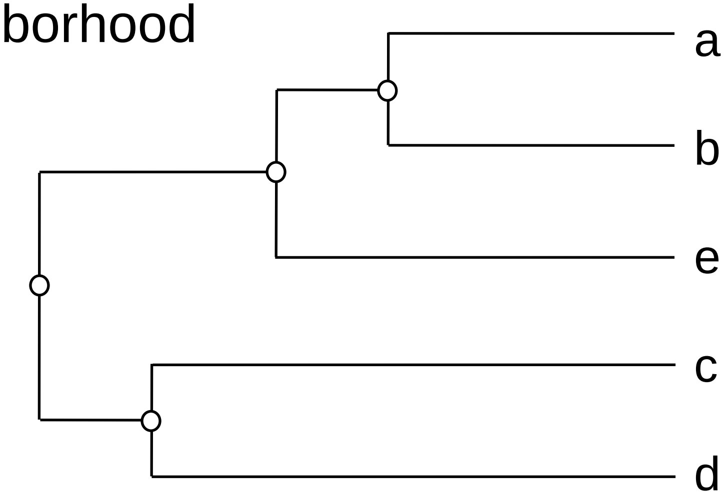
Universität Stuttgart

www.visus.uni-stuttgart.de

Vertex Clustering

- Hierarchical clustering (bottom-up) approach with average linkage criteria
- The *Jaccard coefficient* is used to compute the similarity between two vertices v_i and v_j based on common neighborhood

$$J_w(\overline{V_i}, \overline{V_j}) = \frac{\sum_{u \in \overline{V_i} \cap \overline{V_j}} W(u)}{\sum_{u \in \overline{V_i} \cup \overline{V_j}} W(u)} \in [0, 1]$$



Hierarchical clustering (bottom-up) example

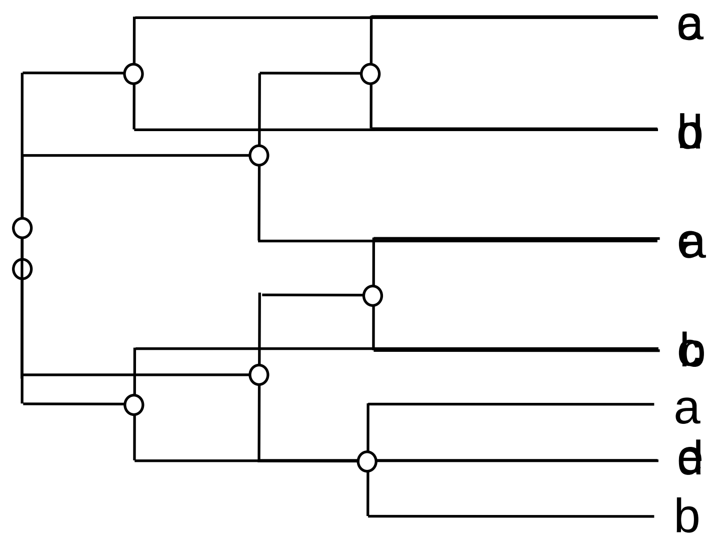
- $\overline{V_i}$ and $\overline{V_j}$ are sets of direct neighbors for vertices v_i and v_j, respectively
- $W(u) := w(u, v_i) + w(v_i, u) + w(u, v_j) + w(v_j, u)$

Universität Stuttgart

www.visus.uni-stuttgart.de

Vertex Reordering

- Finding a better arrangement of vertices while preserving the hierarchical organization (successive augmentation approach [Sil98])
  - The cluster tree is top-down traversed
  - Swap the left and right children and compute the edge-crossing cost
  - Keep the arrangement that achieves the minimum cost
  - The edge cost is relative to the length of the edge and weighted by the frequency of that edge

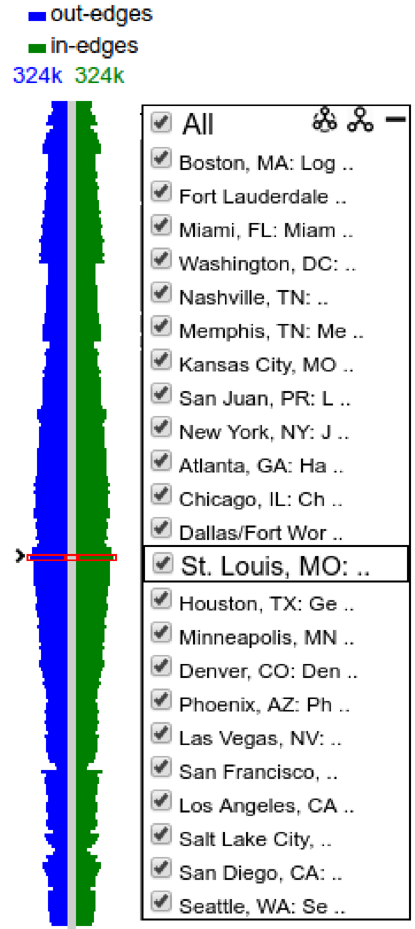


Vertex reordering example

Universität Stuttgart

Interaction Techniques

- Interactive Filtering
  - Vertices
  - Edge weight
  - Densities
  - Time
  - Spatio-temporal



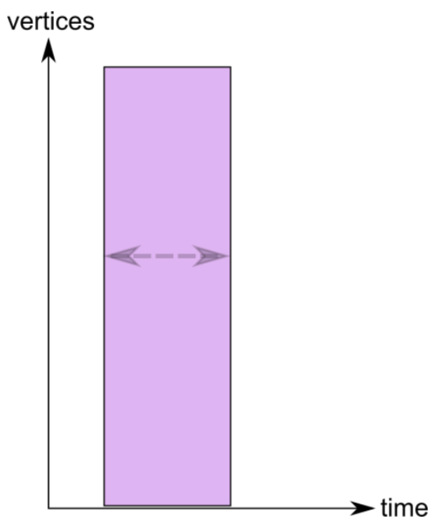
The vertices side menu

www.visus.uni-stuttgart.de

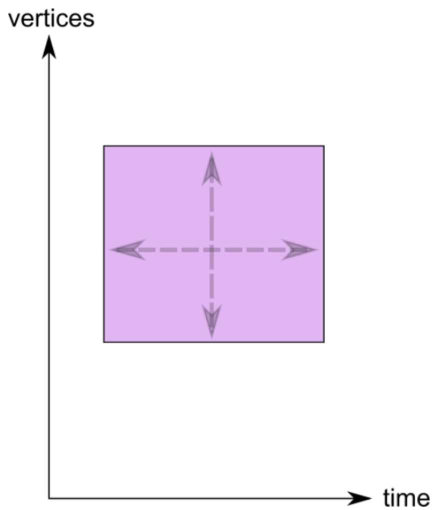
www.visus.uni-stuttgart.de

Interaction Techniques

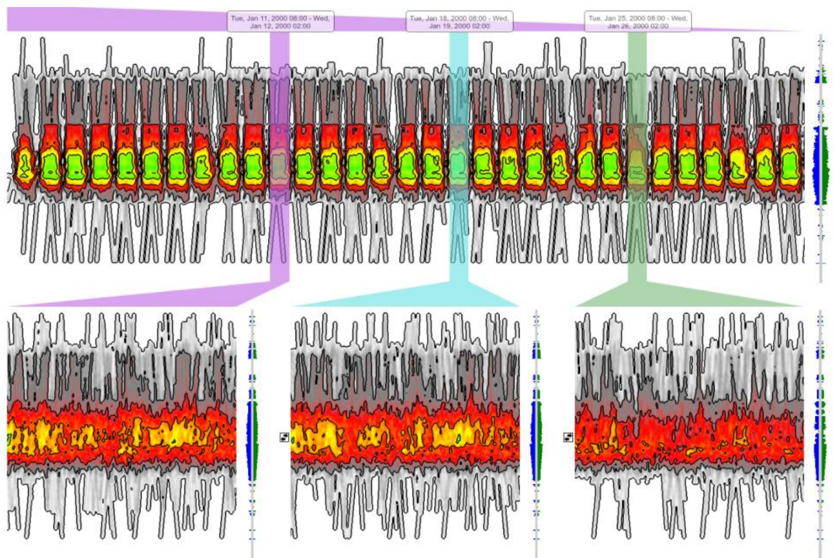
- Selection
  - Time selection
  - Space-time selection
  - Multiple selections



Time selection



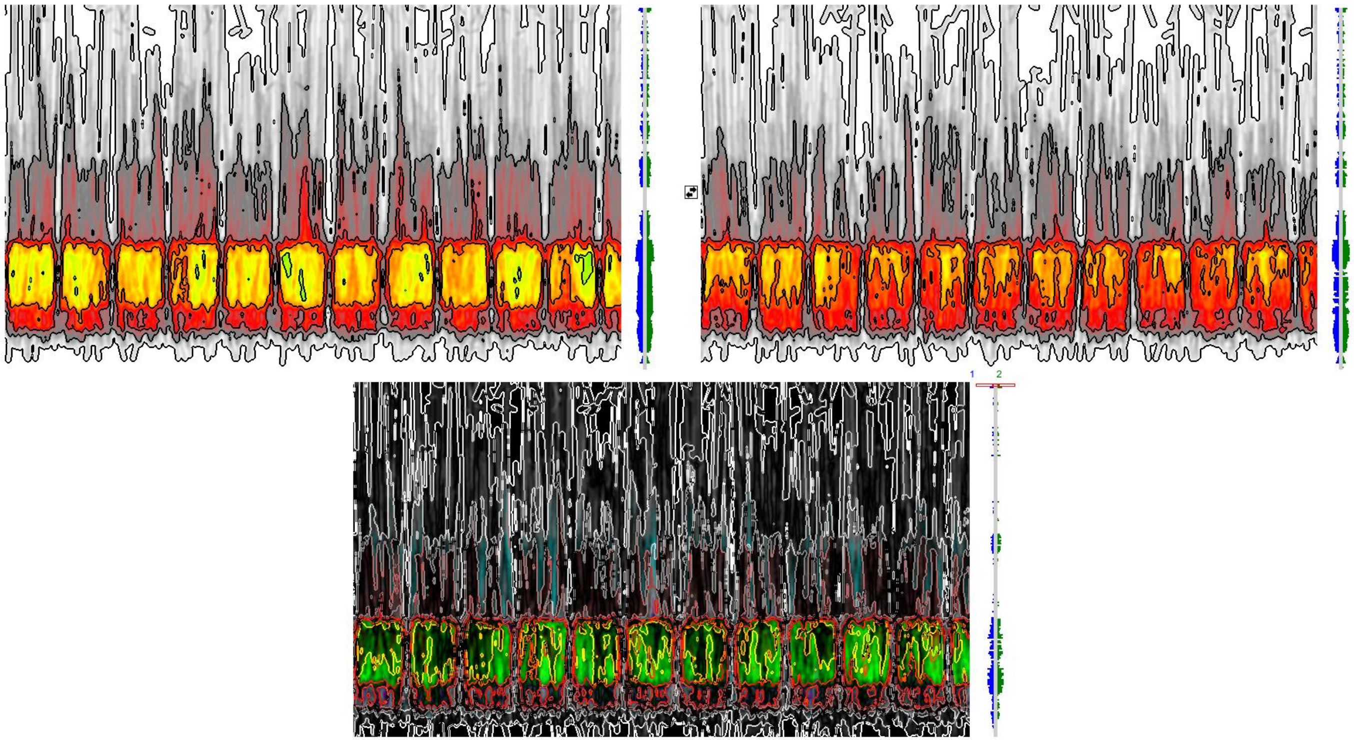
Space-time selection



Multiple selections on the same time period

Interaction Techniques

- Algorithmic Comparison



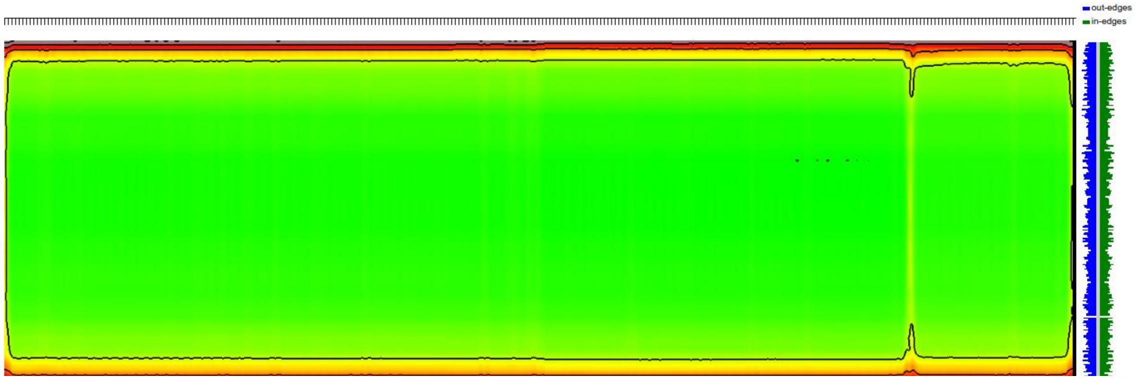
Algorithmic comparison between two timescales

Application Example

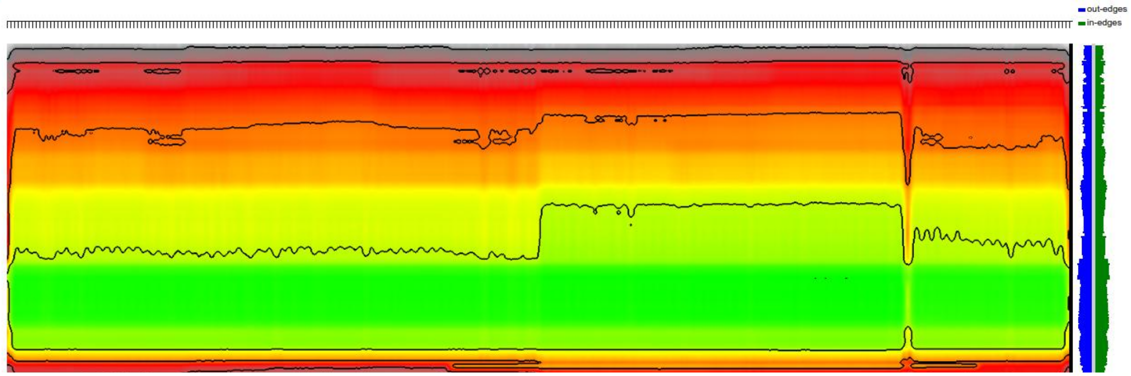
- US domestic flight traffic dataset
- Two years of flight data (from January 1st, 2000 to December 31st, 2001)
- The data is available on a per-minute basis
- 234 vertices (airports)
- ~10 million weighted edges (flight connections)
- 1,051,200 time steps on a per-minute basis

Vertex Clustering and Reordering

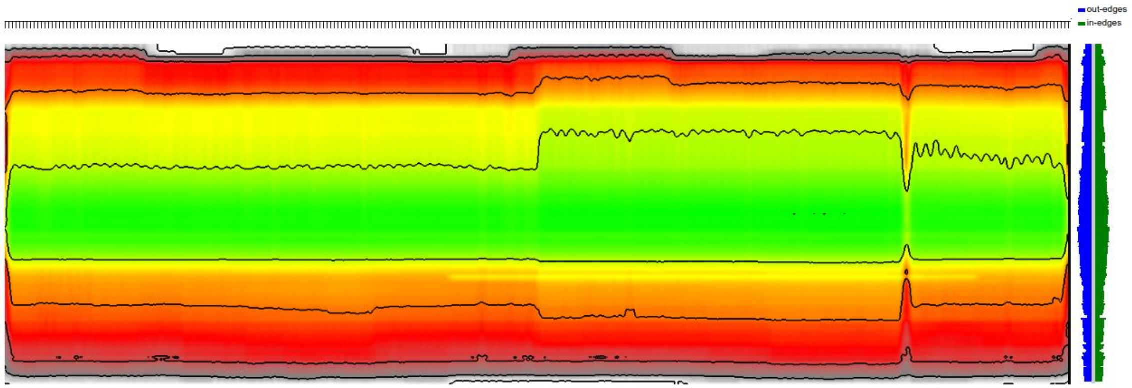
Raw



Clustered



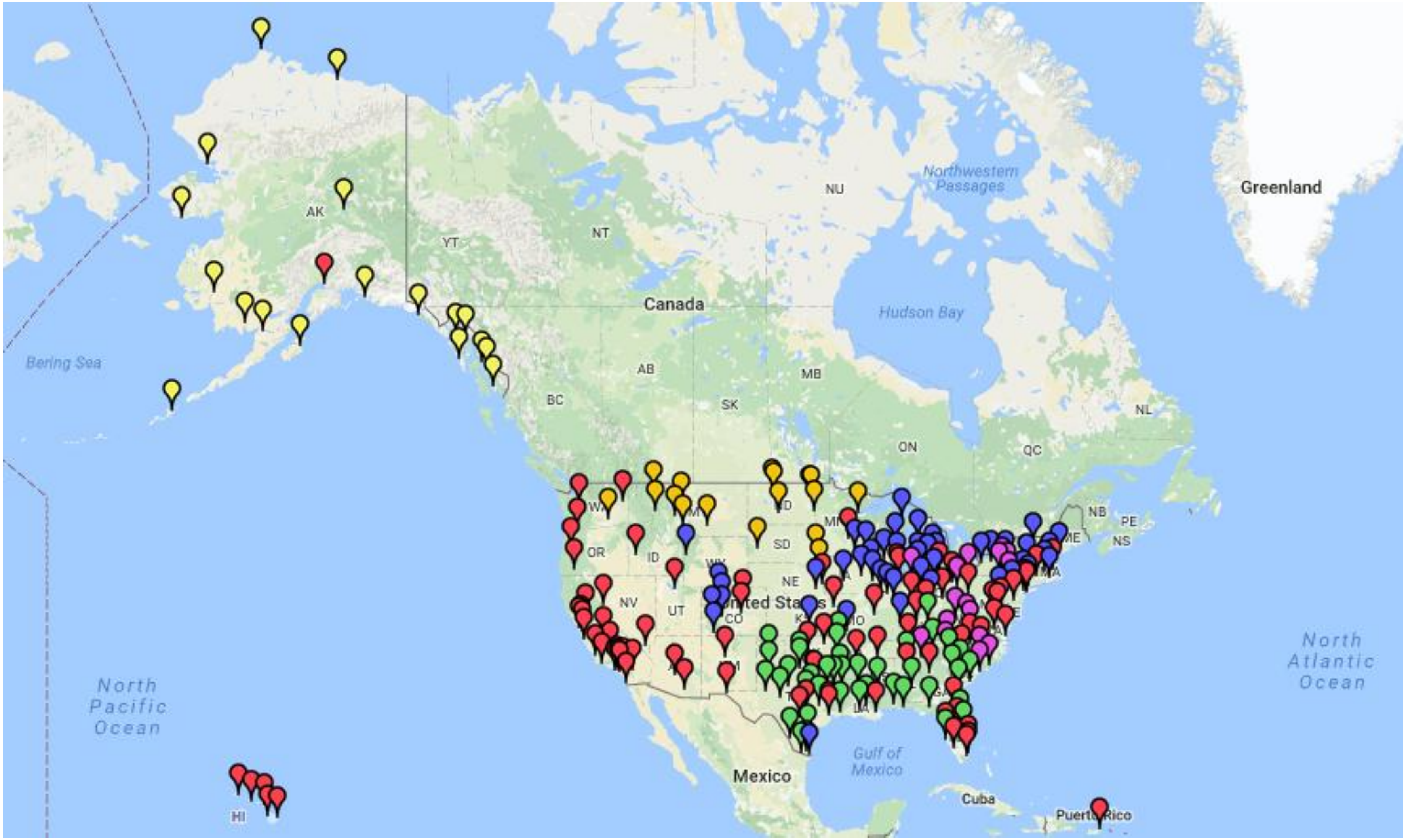
Reordered



Universität Stuttgart

www.visus.uni-stuttgart.de

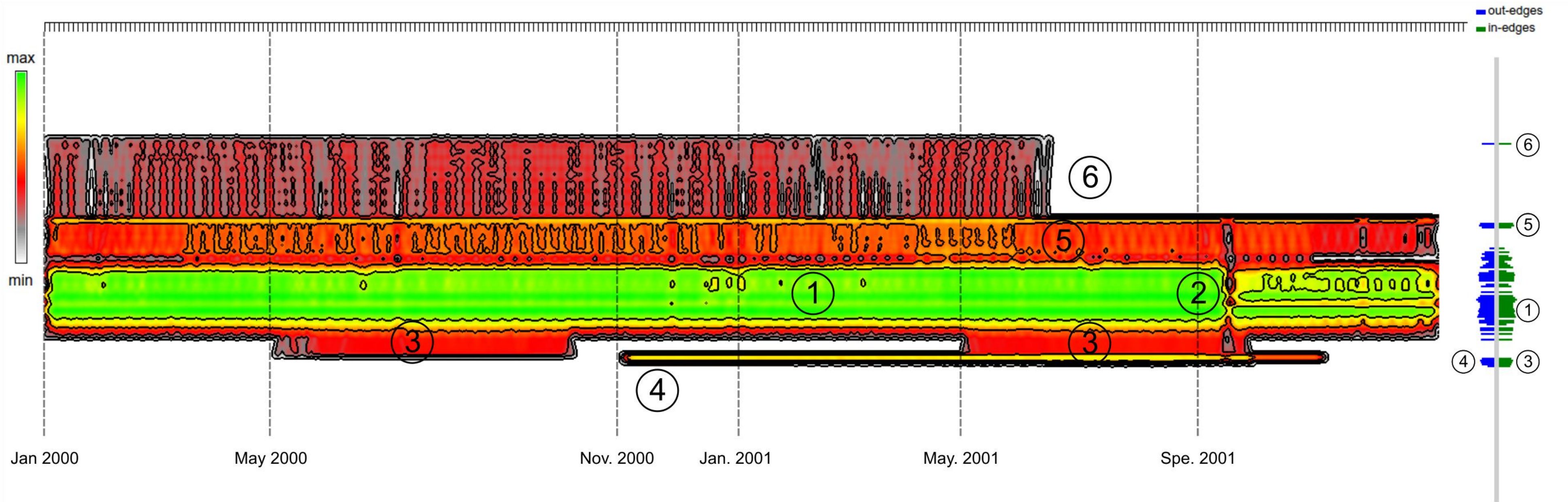
Vertex Clustering and Reordering



Clusters visualized on the map of the United States

www.visus.uni-stuttgart.de

Identifying Spatio-Temporal Patterns



Filtering at threshold value 18 reveals some interesting patterns

(1) Los Angeles Int., Phoenix Sky Harbor Int., Chicago O'Hare Int., Dallas/Fort Worth Int

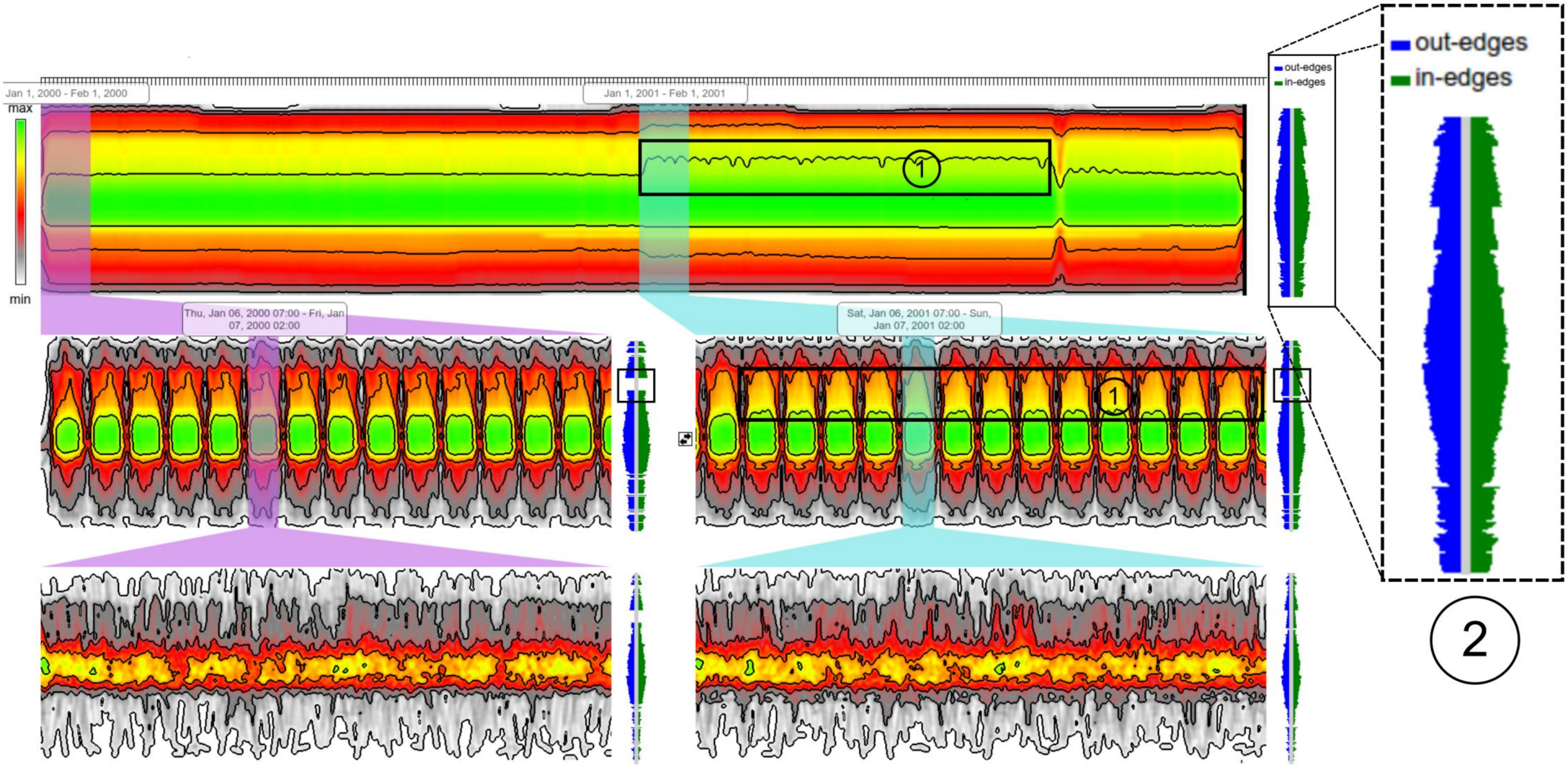
(3) Ted Stevens Anchorage ✈ Seattle/Tacoma Int.

(5) Dallas Love Field, William P Hobby ✈ Austin – Bergstrom Int., Louis Armstrong NOLA

(6) Theodore Francis Green State ✈ Baltimore/Washington

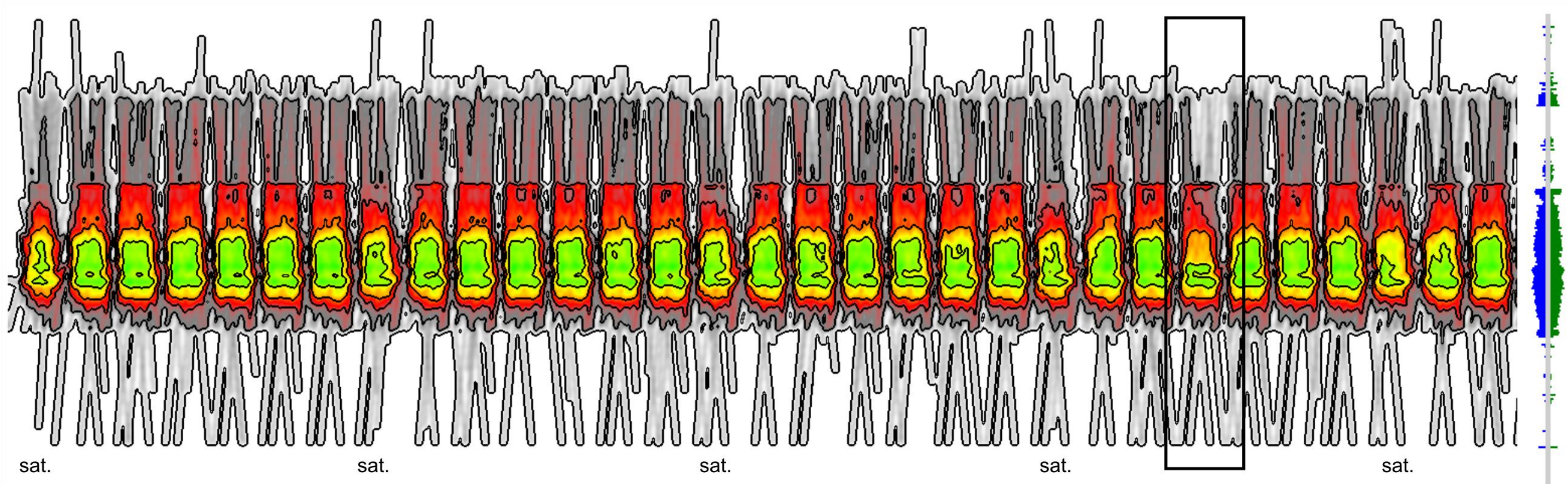
(4) Kona, Lihue, Honolulu, Kahului

Multiscale View



The dynamic graph on multiple timescales

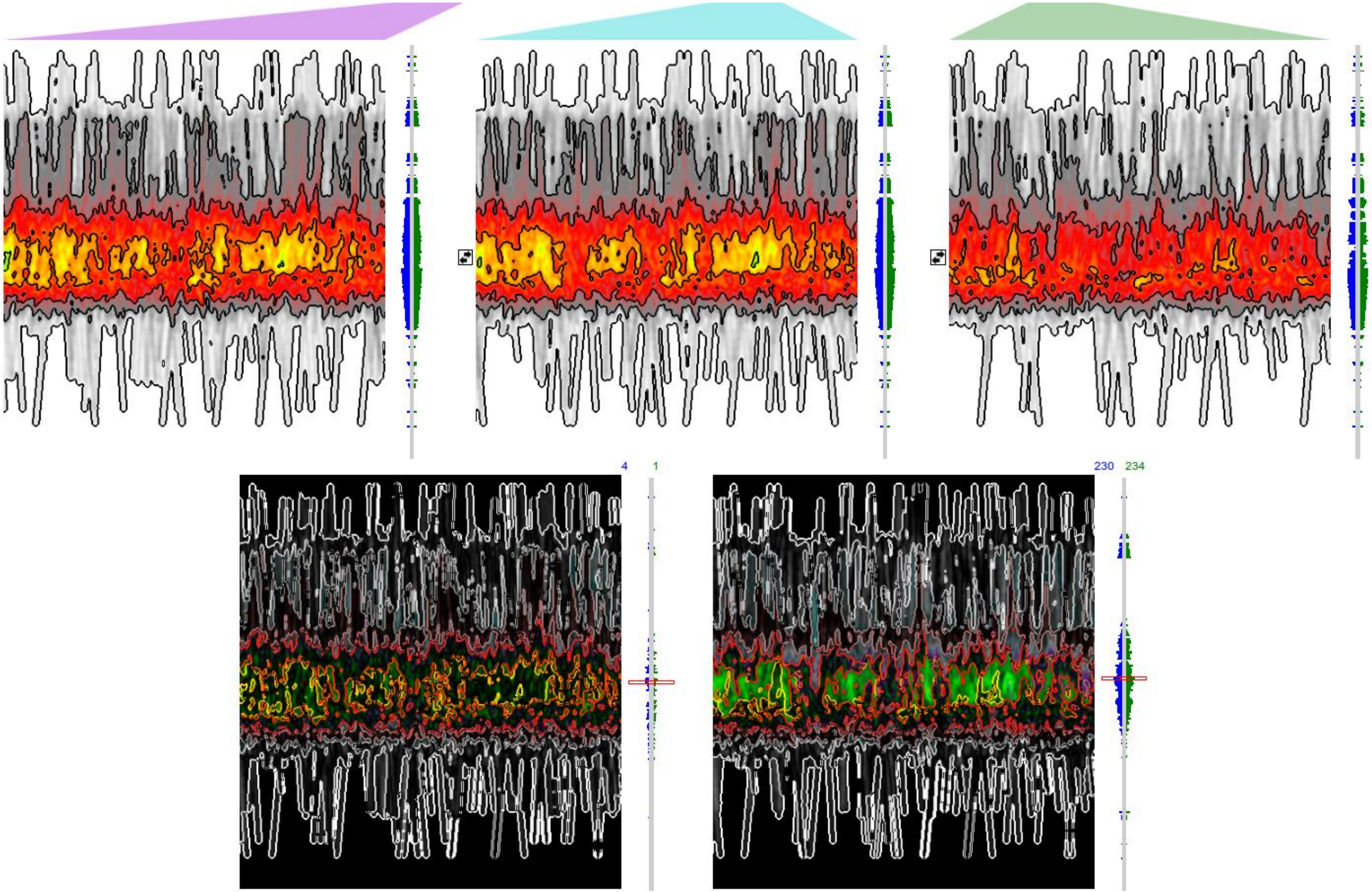
Multiscale View



Filtered flight data for January in 2000 with an unusual pattern behavior on 25th of January

Newark Liberty Int.
Philadelphia Int.
LaGuardia

Algorithmic Comparison



Algorithmic comparison of three selected Tuesdays (11th , 18th and 25th) of January 2000 respectively.

www.visus.uni-stuttgart.de

Discussion and Limitations

- The bipartite layout lives in a one-dimensional space
- The time period on which the clustering and ordering is based
- The algorithm parameters should be chosen carefully
- A more general comparison approach would be more suitable, but also more time-complex
- Other interaction techniques should be added in the future

Conclusion

- A visualization approach to provide an overview on multiple time scales of the time-varying relational data
- Our Contribution
  - Compact visualization design (top-down and side-by-side stacking)
  - Reducing the amount of visual clutter (clustering and ordering techniques)
  - Set of interaction techniques
- An application example demonstrates the applicability of our approach
- Future Work
  - Automatically generates a default setting for the multiple timescales
  - Trying other artificial and real-world dynamic datasets
  - Comparing with other visualization approaches
  - Implementing additional interaction techniques

Universität Stuttgart

References

- C. S. Collberg, S. G. Kobourov, J. Nagra, J. Pitts, K. Wampler. “A System for Graph-Based Visualization of the Evolution of Software.” In: *Proceedings ACM 2003 Symposium on Software Visualization*. 2003, pp. 77–86, 212– 213 (cit. on p. 22).
- B. Bach, E. Pietriga, J.-D. Fekete. “Visualizing Dynamic Networks with Matrix Cubes.” In: *CHI Conference on Human Factors in Computing Systems*. 2014, pp. 877–886 (cit. on p. 23).
- B. Bach, N. H. Riche, T. Dwyer, T. M. Madhyastha, J. Fekete, T. J. Grabowski. “Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks.” In: *Computer Graphics Forum* 34.3 (2015), pp. 31–40 (cit. on pp. 23, 24).
- M. Burch, S. Diehl. “TimeRadarTrees: Visualizing Dynamic Compound Digraphs.” In: *Computer Graphics Forum* 27.3 (2008), pp. 823–830 (cit. on pp. 25, 26).
- S. van den Elzen, D. Holten, J. Blaas, J. J. van Wijk. “Dynamic network visualization with extended massive sequence views.” In: *IEEE transactions on visualization and computer graphics* 20.8 (2014), pp. 1087–1099 (cit. on pp. 25, 26, 40).
- S. van den Elzen, D. Holten, J. Blaas, J. J. van Wijk. “Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration.” In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 1–10 (cit. on pp. 25, 26).
- M. Burch, C. Vehlow, F. Beck, S. Diehl, D. Weiskopf. “Parallel Edge Splatting for Scalable Dynamic Graph Visualization.” In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2344–2353 (cit. on pp. 26, 28).
- M. Burch, M. Hlawatsch, D. Weiskopf. “Visualizing a Sequence of a Thousand Graphs (Or Even More).” In: *Computer Graphics Forum* (2017) (cit. on pp. 17, 18, 27, 28, 40).

References

- C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004 (cit. on pp. 22, 28, 44, 47, 64).
- J. P. i Silvestre. “Approximation Heuristics and Benchmarkings for the MinLA Problem.” In: *Proceedings of Algorithms and Experiments (ALEX)*.1998, pp. 112–128 (cit. on p. 34)
- U. S. D. of Transportation. United States Department of Transportation. https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time. (Accessed on 07/23/2017). July 2017 (cit. on pp. 57, 67, 69, 72).